

APD - Lab2

Tuesday, October 09, 2007
12:06 PM

impartirea iteratiilor in for

```
#pragma omp for schedule (static,5)
    se dau 5 iteratii la fiecare thread; daca sunt mai multe se reincepte
    deci impartirea se face egal
#pragma omp for schedule (dynamic,5)
    fiecare isi ia un nou set dupa ce a terminat
#pragma omp for schedule (guided,5)
    daca raman putine iteratii de executat la sfarsit, ele se vor imparti intre threaduri
```

#pragma omp parallel for
scriere rapida a doua directive: parallel si for

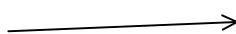
```
#pragma omp parallel
{
    #pragma omp sections
    {
        //cod executat de primul thread

        #pragma omp section
        //alt thread
        #pragma omp section
        //alt thread
    }
}

sections trebuie sa intre intr-o directiva parallel
```

Variabile:

```
int a,b=5;
#pragma omp parallel
    int x;
```



variabilele a si b sunt shared (partajate), adica toate threadurile vor avea acces la ele
variabila x este variabila privata (private), adica se creeaza cate o variabila x pentru fiecare thread.

#pragma omp thread private(a,b)

- acum, cand vom crea o regiune paralela in program, variabilele aceste vor deveni private si se vor initializa cu valoarea pe care o aveau inainte de regiunea paralela
- directiva va avea efect pentru toate regiunile programului

a=5;

#pragma omp parallel private(a,b)

variabilele au valori reziduale, ca si cum au fost declarate doar

#pragma omp parallel firstprivate(a,b)

variabilele se initializeaza cu valorile pe care le aveau inainte

#pragma omp parallel lastprivate(a,b)

variabilele nu se initializeaza, dar la sfarsit se pastreaza valoarea pe care o aveau la ultima iteratie

#pragma omp parallel shared(x)

daca variabila e declarata in interiorul regiunii paralele, va pastra o singura copie a ei si o va face privata

#pragma omp flush

se executa automat la inceputul si sfarsitul regiunilor paralele

#pragma omp single

cod executat de un singur thread, o singura data
critical este executata de toate threadurile, dar pe rand

#pragma omp master

executat doar de threadul master, cel care a creat initial regiunea paralela, cu id-ul 0

Variabile Lock

un fel de variabila booleana, firele o pot bloca sau debloca

```
tip de date: omp_lock_t zavor;
omp_set_lock(&zavor);
    //secventa critica de cod
omp_unset_lock(&zavor);
omp_destroy_lock(&zavor);
```

Programel mic:)

```
a00x0+a01x1+...+an-1xn-1=b0
a11x1+...+a1,n-1xn-1=b1
.....
an-2,n-2xn-2+an-1,n-1xn-1=bn-2
an-1,n-1xn-1=bn-1
```

#pragma omp parallel

```
for(i=n-1;i>=0;i--)
{
    #pragma omp single
        x[i]=
    #pragma omp for
```

#pragma omp parallel

```
{
    #pragma omp single
        x[i]=
    #pragma omp parallel for
    for{
```